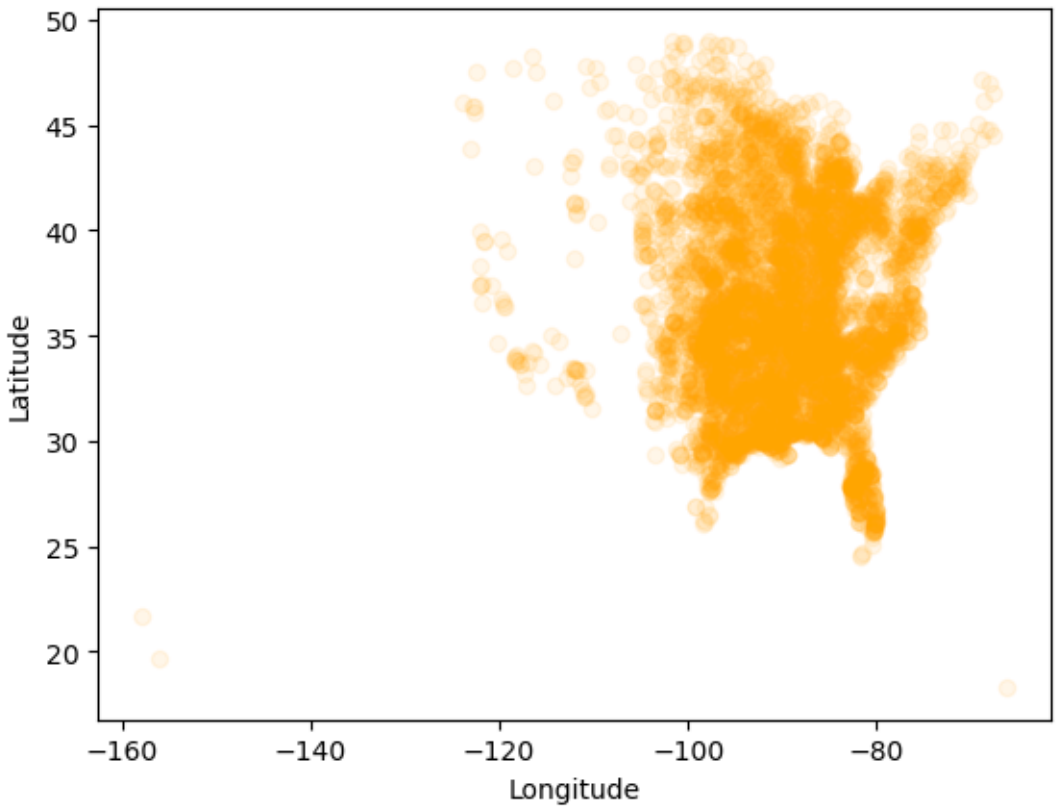
# Lab06

March 16, 2025

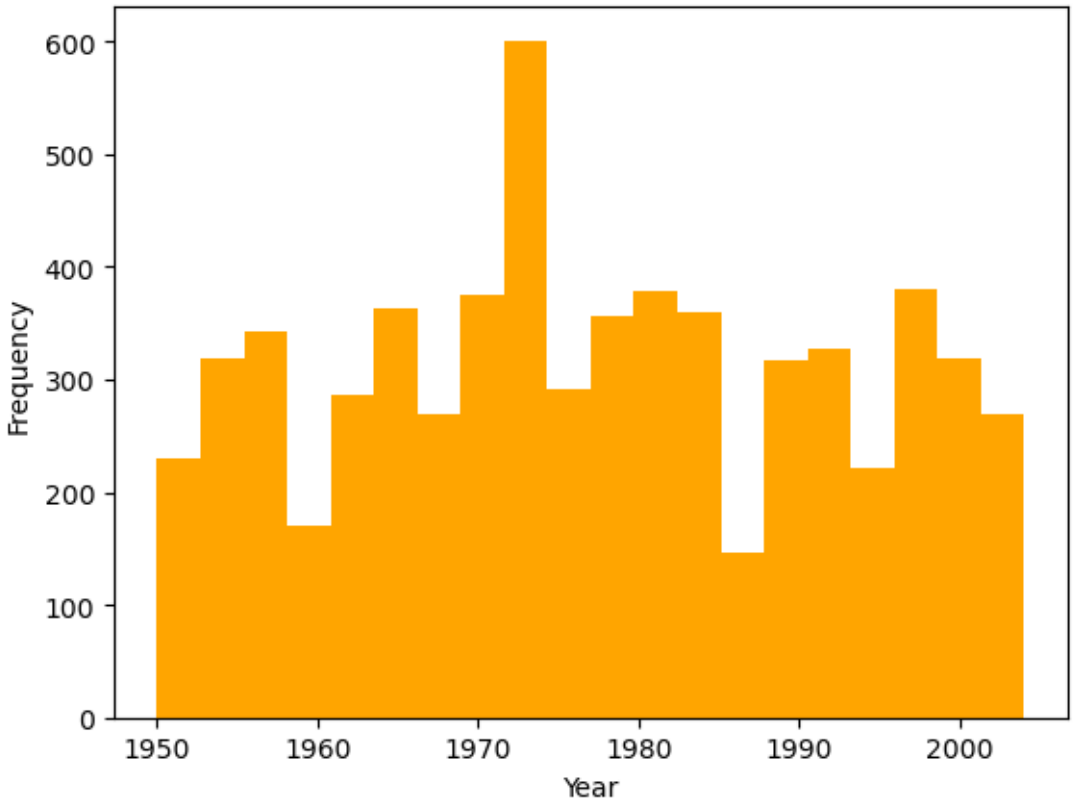
[80]:
```
##imports
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import contextily as ctx
```

[81]:
```
#3. New data frame
torn = pd.read_csv("https://raw.githubusercontent.com/mhaffner/data/master/torn.
 ↪csv")

torn_filtered = torn[torn["INJ"] > 0][["FATAL", "YEAR", "INJ", "F_SCALE", "x",␣
 ↪"y"]]
```

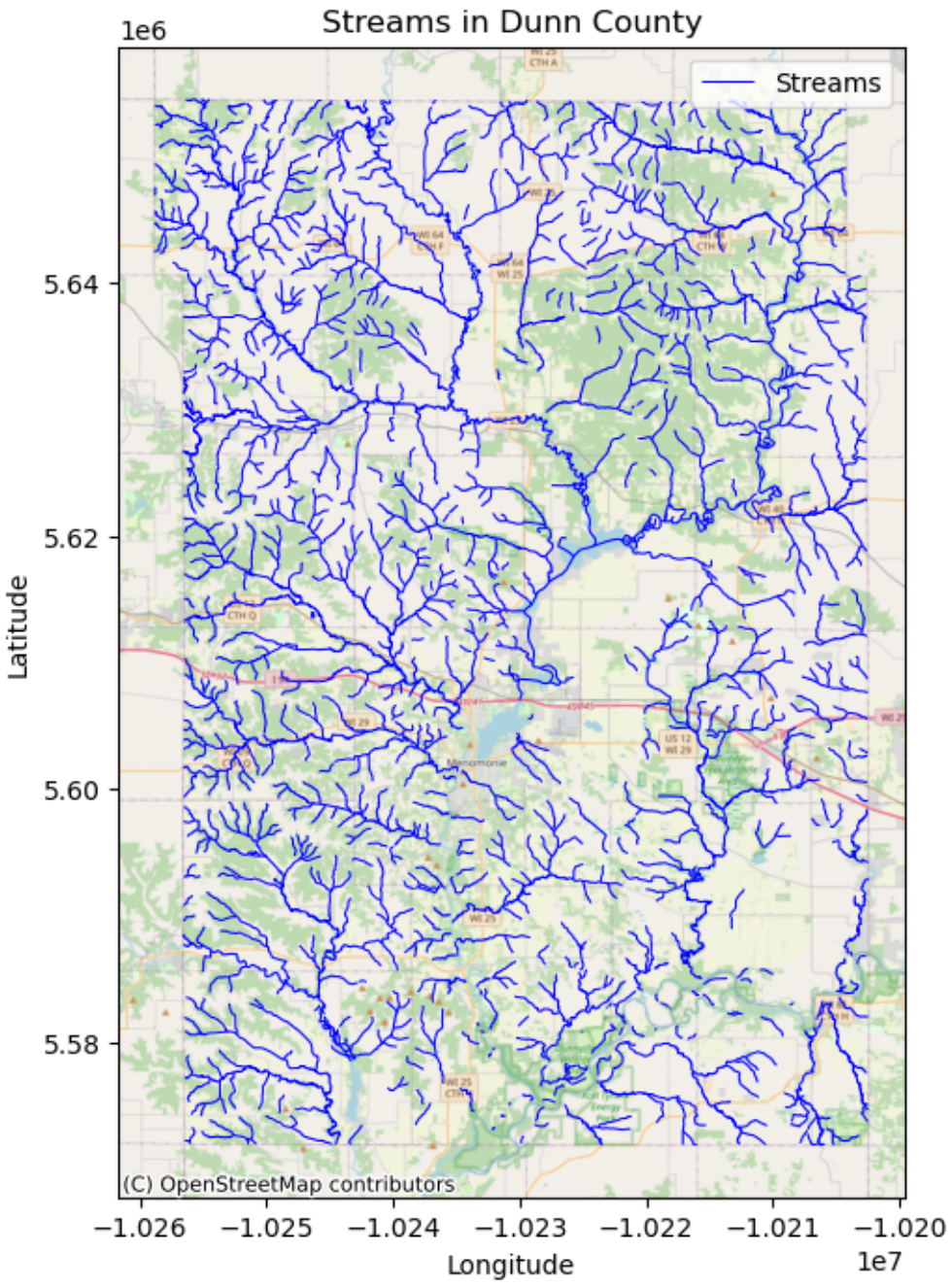
[82]:
```
#4. Scatterplot
plt.scatter(torn_filtered["x"], torn_filtered["y"], c="orange", alpha=0.09)
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```




[83]:
```
#5. Histogram
plt.hist(torn_filtered["YEAR"], bins=20, color="orange")
plt.xlabel("Year")
plt.ylabel("Frequency")
plt.show()
```

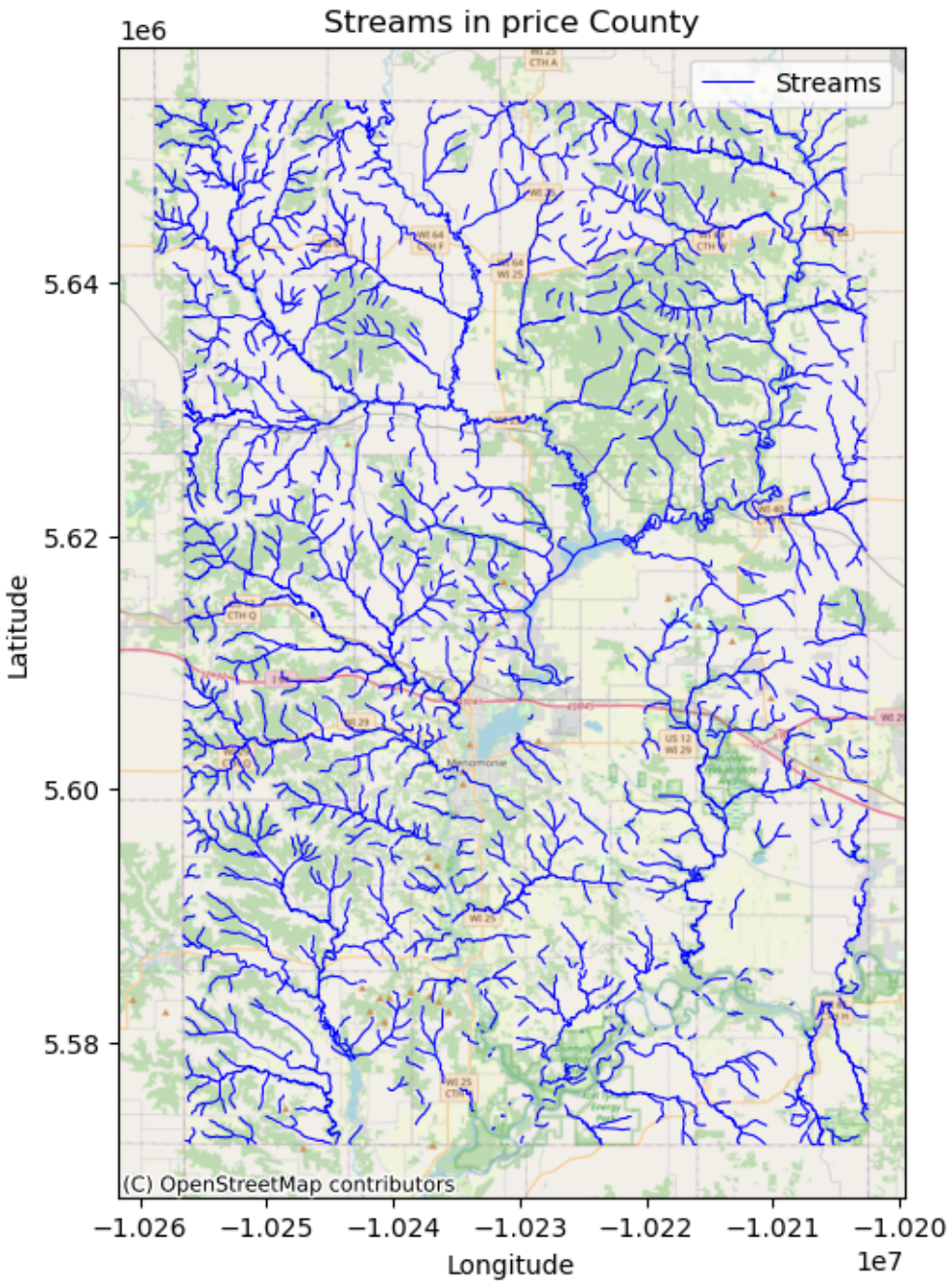



[84]:
```
#6 - 8. Creating Dunn Streams Plot
streams = gpd.read_file("https://gitlab.com/mhaffner/data/-/raw/master/
 ↪dunn-county-streams.geojson")

# Changeing the CRS
streams = streams.to_crs(epsg=3857)

#plotting streams
fig, ax = plt.subplots(figsize=(10, 8))
streams.plot(ax=ax, linewidth=0.75, color="blue", label="Streams")

# Basetiles
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)

# Customize the plot
ax.set_title("Streams in Dunn County")
ax.set_xlabel("Longitude")
ax.set_ylabel("Latitude")
plt.legend()
plt.show()
```



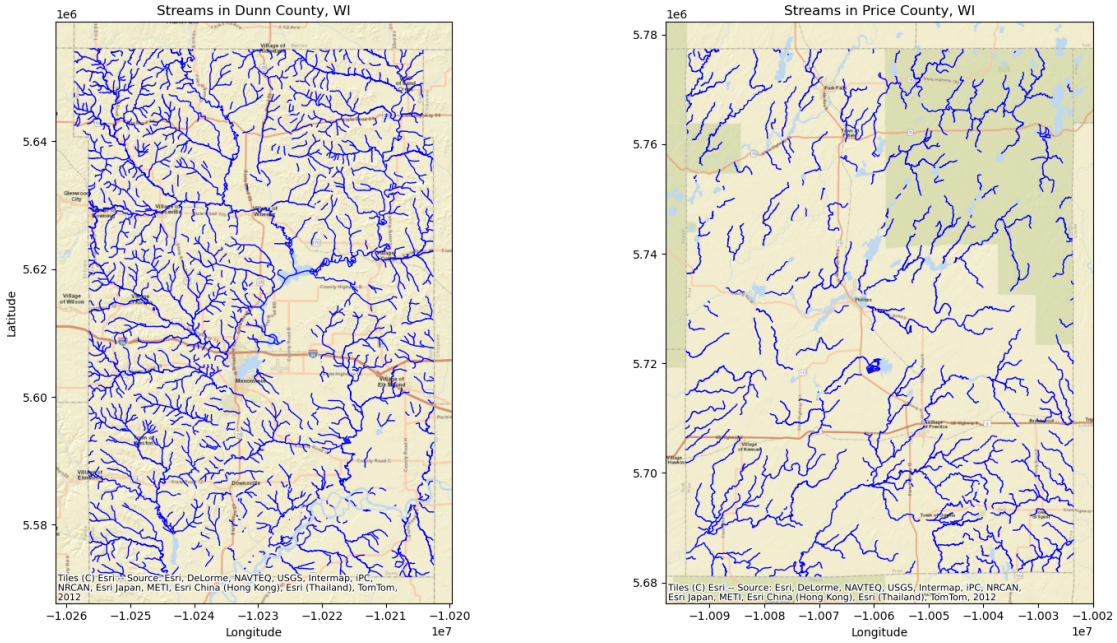

[85]:
```
#6. Read CSV with geopandas
streams_price = gpd.read_file("https://gitlab.com/mhaffner/data/-/raw/master/
 ↪price-county-streams.geojson")

# Changeing the CRS
streams_price = streams_price.to_crs(epsg=3857)
```

```
#plotting streams
fig, ax = plt.subplots(figsize=(10, 8))
streams.plot(ax=ax, linewidth=0.75, color="blue", label="Streams")

# Basetiles
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)

# Customize the plot
ax.set_title("Streams in price County")
ax.set_xlabel("Longitude")
ax.set_ylabel("Latitude")
plt.legend()
plt.show()
```



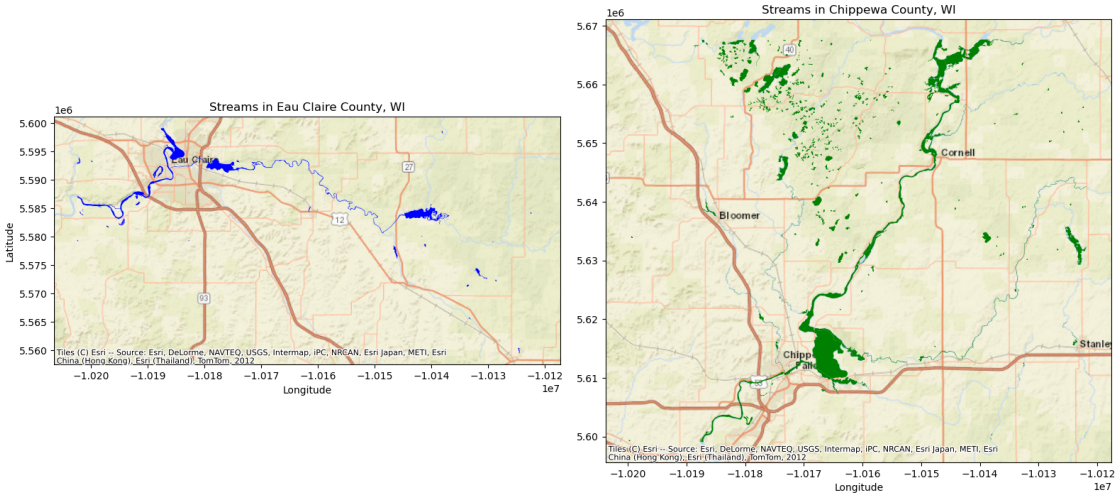

[86]:
```
#10. Subplots to plot side-by-side
fig, axes = plt.subplots(1, 2, figsize=(16, 8))

# Plot Dunn County
streams_dunn.plot(ax=axes[0], linewidth=1, color="blue", label="Dunn Streams")
ctx.add_basemap(axes[0], source=ctx.providers.Esri.WorldStreetMap)  #Used Esri␣
 ↪instead, Having troubles with Stamen
```

```
axes[0].set_title("Streams in Dunn County, WI")
axes[0].set_xlabel("Longitude")
axes[0].set_ylabel("Latitude")

# Plot Price County
streams_price.plot(ax=axes[1], linewidth=1, color="blue", label="Price Streams")
ctx.add_basemap(axes[1], source=ctx.providers.Esri.WorldStreetMap)
axes[1].set_title("Streams in Price County, WI")
axes[1].set_xlabel("Longitude")

plt.tight_layout()
plt.show()
```




[91]:
```
eau_claire_url = "https://www2.census.gov/geo/tiger/TIGER2023/AREAWATER/
 ↪tl_2023_55035_areawater.zip"
chippewa_url = "https://www2.census.gov/geo/tiger/TIGER2023/AREAWATER/
 ↪tl_2023_55017_areawater.zip"

# Load datasets
streams_eau_claire = gpd.read_file(eau_claire_url)
streams_chippewa = gpd.read_file(chippewa_url)

# Change EPSG
streams_eau_claire = streams_eau_claire.to_crs(epsg=3857)
streams_chippewa = streams_chippewa.to_crs(epsg=3857)
```

```
# Create side-by-side subplots
fig, axes = plt.subplots(1, 2, figsize=(16, 8))

# Plot Eau Claire County
streams_eau_claire.plot(ax=axes[0], linewidth=1, color="blue", label="Eau␣
 ↪Claire Streams")
ctx.add_basemap(axes[0], source=ctx.providers.Esri.WorldStreetMap)
axes[0].set_title("Streams in Eau Claire County, WI")
axes[0].set_xlabel("Longitude")
axes[0].set_ylabel("Latitude")

# Plot Chippewa County
streams_chippewa.plot(ax=axes[1], linewidth=1, color="green", label="Chippewa␣
 ↪Streams")
ctx.add_basemap(axes[1], source=ctx.providers.Esri.WorldStreetMap)
axes[1].set_title("Streams in Chippewa County, WI")
axes[1].set_xlabel("Longitude")

plt.tight_layout()
plt.show()
```




[ ]: